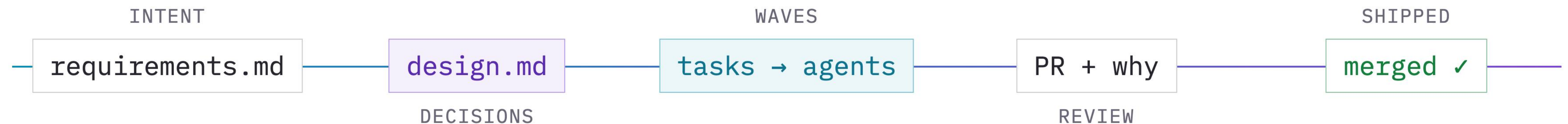


Software Factories:

Reaching Terminal Velocity

From /plan to full lifecycle automation: what it taught us, and how the same machine can run a company.



Burak Özafsar

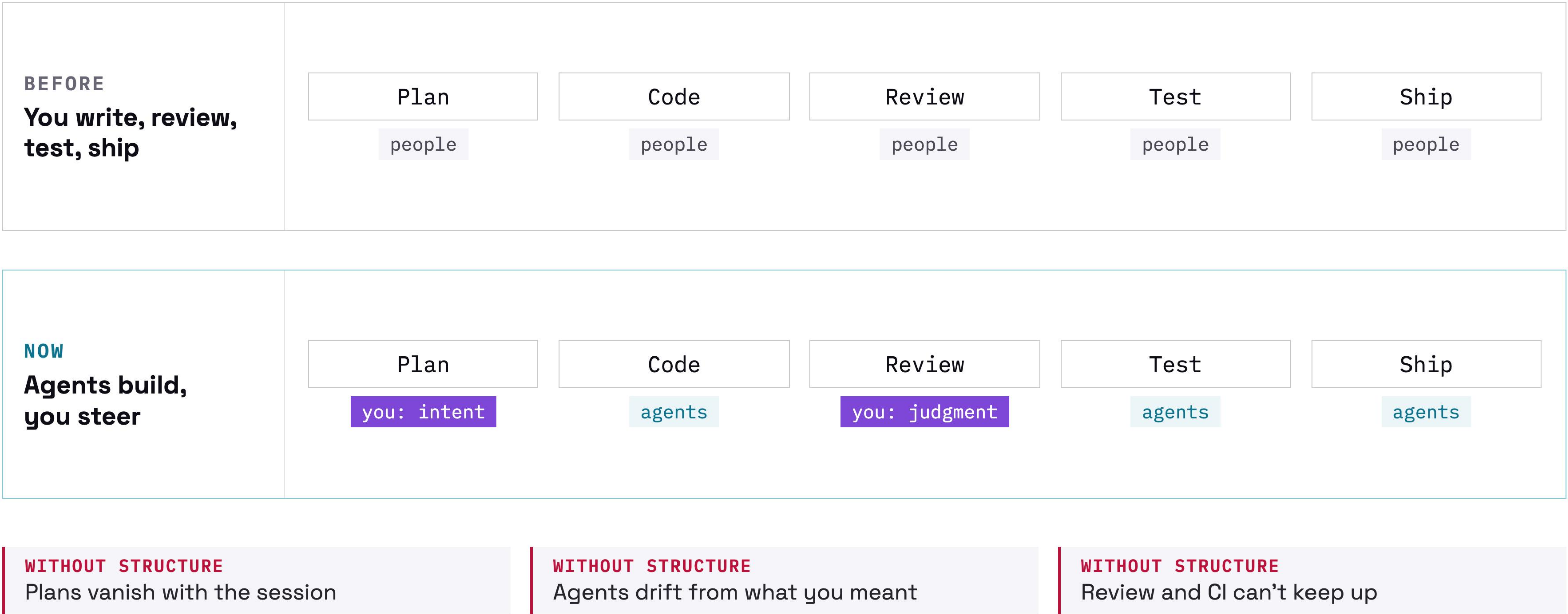
CTO • MUTAGENT

Building the Agent Factory at Mutagent.

Claude Meetup: Claude For Founders

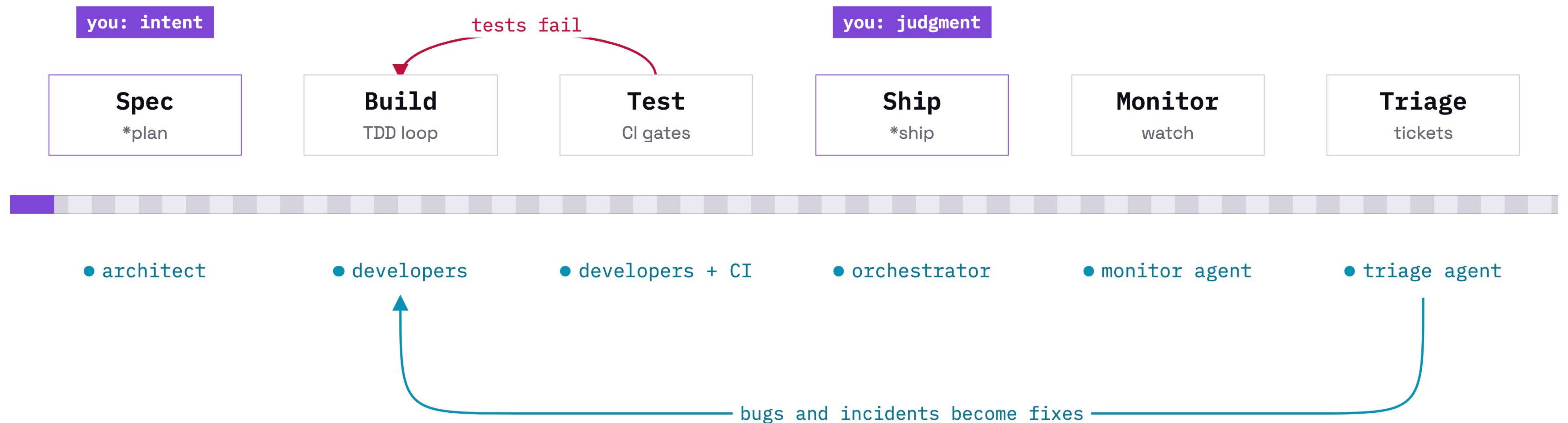
28 September 2026

The development lifecycle is changing



A software factory runs the whole lifecycle

A team of agents with **fixed roles**, **shared rules** and **repeatable workflows**. Intent goes in, reviewed and shipped changes come out, with a record at every step.



Maps, rules and memory answer different questions

CONTEXT MAP

Where things are

Base map always loaded

LOADED	Base map always, details on demand
WRITTEN BY	The architect, when structure changes
CHANGES	With the repo layout

ENGINEERING RULES

Rules to follow

Always loaded

LOADED	In every session, from the start
WRITTEN BY	You, deliberately
CHANGES	Rarely, by review

MEMORY

What we learned

Grows every session

LOADED	By relevance, from an index
WRITTEN BY	Agents, from feedback and incidents
CHANGES	Every session

```
.memory/context-map.md
### Product
- [product context](product-context.md)
- [architecture](system-architecture.md)
### Repositories
- [server](server/.../repository-context.md)
- [ui](ui/.../repository-context.md)

OKF-style references and indexing
```

```
graph LR
    PM1[PRODUCT MEMORY  
product-context.md] --- CM[context-map.md]
    PM2[PRODUCT MEMORY  
system-architecture.md] --- CM
    GR[GOVERNANCE RULE  
system-design-rules.md] --- CM
    CM --- RC1[REPOSITORY CONTEXT  
server/repository-context.md]
    CM --- RC2[REPOSITORY CONTEXT  
ui/repository-context.md]
    CM --- RR[REPO RULE  
server/coding-rules.md]
```

One index links product memory, rules and every repository context

Markdown is the new software

① **SYSTEM** the shapes

- agent blueprint
- command blueprint
- skill blueprint

② **DOMAIN** .architech/domains/software, symlinked into every repo's .claude/

- agents/

orchestrator · architect · developer · reviewer
- skills/

playwright-cli · agent-browser · ctx7-docs
- commands/

*plan · *dispatch · *ship
- workflows/

static and dynamic

③ **CONTEXT** the codebase the factory works on

PRODUCT LEVEL

- .memory/

context-map.md

product-context.md

system-architecture.md

tech-stack.md

.engineeringrules/

GOVERNANCE

system-design-rules.md

documentation-rules.md
- REPO LEVEL · SAME SHAPE IN EVERY REPO
- server/

.memory/

repository-context.md

.engineeringrules/

coding · testing

ui/

.memory/

repository-context.md

.engineeringrules/

coding · design

cli/

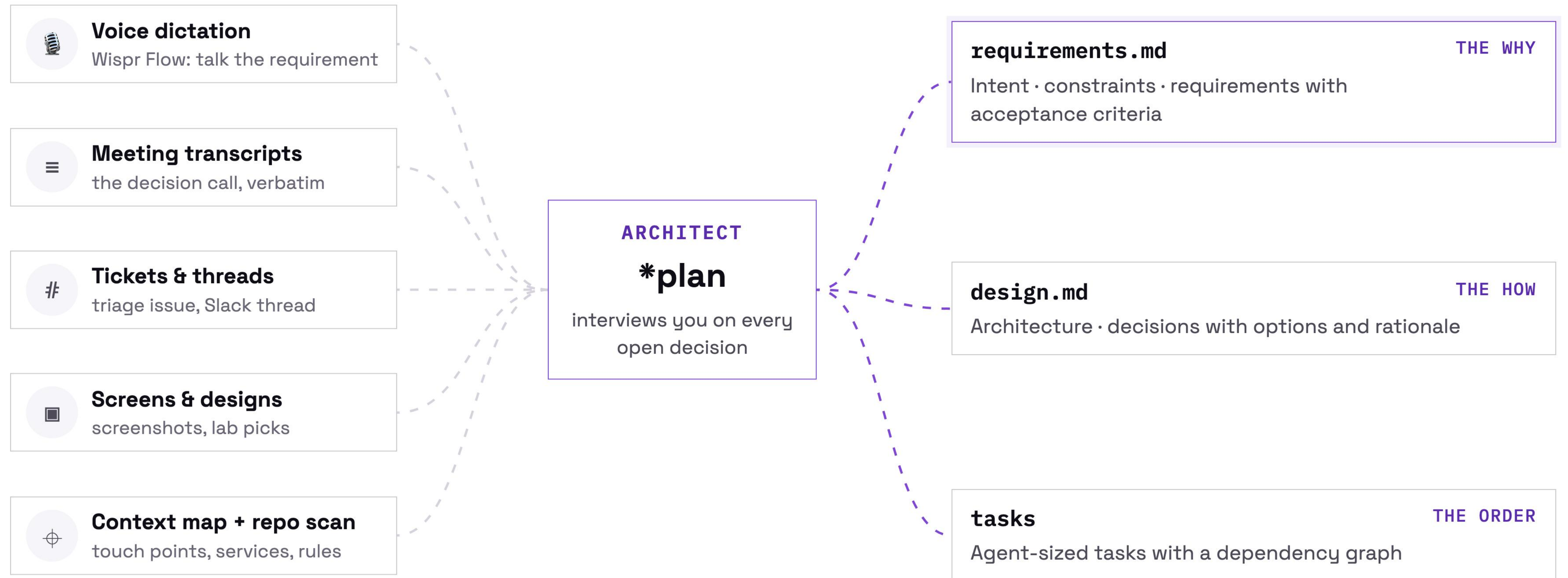
.memory/

repository-context.md

.engineeringrules/

coding · testing
- Software Factories: Reaching Terminal Velocity
- 05 / 15

A plan starts as talk and ends as three files



A real plan: requirements, design, tasks

● ● ●

claude.ai/artifact/... · Add sandbox providers to Mutagent: PRD

BACKEND PRD

I Requirements

II Design

III Tasks

PART I

§0 Intent

§2 Key problems

§3 Functional req.

§4 Non-functional

PART II

§9 Architecture

§12 Decisions

§15 Data model

PART III

Wave 0 ... 4

Verification

NEEDS YOUR DECISION · new tables · one UX ruling · retention

§0 Intent

ID	INTENT	WHY IT IS INTENT
INT-1	The core stays built for agents; the web gets a presentation layer on top	Additive-only is a rule; this is a taste that picks between two valid designs
INT-2	One platform, three faces: web, CLI, desktop see the same truth	No single requirement names parity across clients
INT-4	The web never shows invented data as real	A marker rule is UI; the want is honesty

45 tasks

W0

W1

W2

W3

W4

I · REQUIREMENTS

Intent first: the wants no requirement can express. Then problems, and every requirement with acceptance criteria.

II · DESIGN

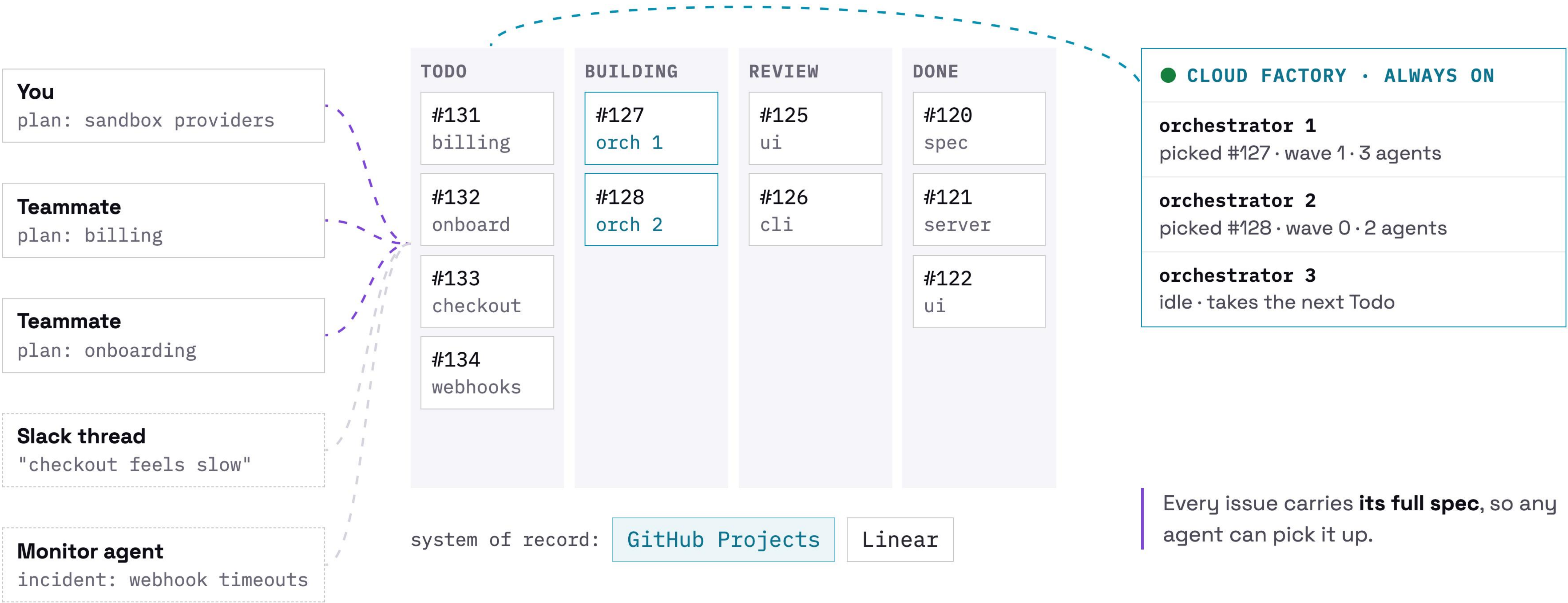
Decisions with their options and **why**. Anything risky, like new tables, is flagged for a human before build.

III · TASKS

45 tasks in 5 waves, owners proposed, and the tasks with no blockers marked to start first.

Open the PRD ↗

Many planners, one board, builders that never stop



Agents build in waves; sessions talk directly

● ● ● orchestrator · claude

> *dispatch #2540

fetched #2540 + 12 sub-issues · waves computed

Wave 0 → spawning team

| developer #2541 catalog lane/2541

| reviewer #2541 queued

| monitor CI + trunk armed

✓ #2541 PR opened · review passed

Wave 1 → #2542 · #2543 · #2544

3 developers in parallel ...

SESSION ↔ SESSION

session: backend → session: web

@web the contract for GET /views/home is on the trunk (#2410). You can swap the mock.

@backend swapped; end-to-end green. PR #2412.

Parallel sessions coordinate with mentions, so you stop being the relay between them.

PRE-COMMIT GUARD · EVERY DEVELOPER AGENT

failing test

 →

implement

 →

test passes

 →

lint · types · build · tests

 →

commit

↺ any check fails: back to implement, nothing broken lands

Written intent lets agents check what you meant

- PR #2410
- PR #2411
- PR #2412
- PR #2413
- PR #2414
- ...

1

Does it work?
acceptance criteria from the spec

2

Is it built our way?
design principles: each decision with its reason

3

Is it what we meant?
intent from the requirements

design-principles.md • per repo

Product, system and architecture decisions, collected over time from decision ledgers, feedback and code reviews.

Repository pattern for data access

why: swap stores in tests

Result types over throwing

why: errors stay typed, never silent

Additive migrations only

why: every rollback stays safe

An agent can tell you the code works. Only written intent tells it whether this was the implementation you wanted.

Every PR says what it read, decided and why

DIFF VS SPEC #120

FR-RP-1 idempotent	✓
FR-RP-3 served sandboxed	✓
FR-RP-5 illegal transition → 409	✗
↩ back to build, then re-review	

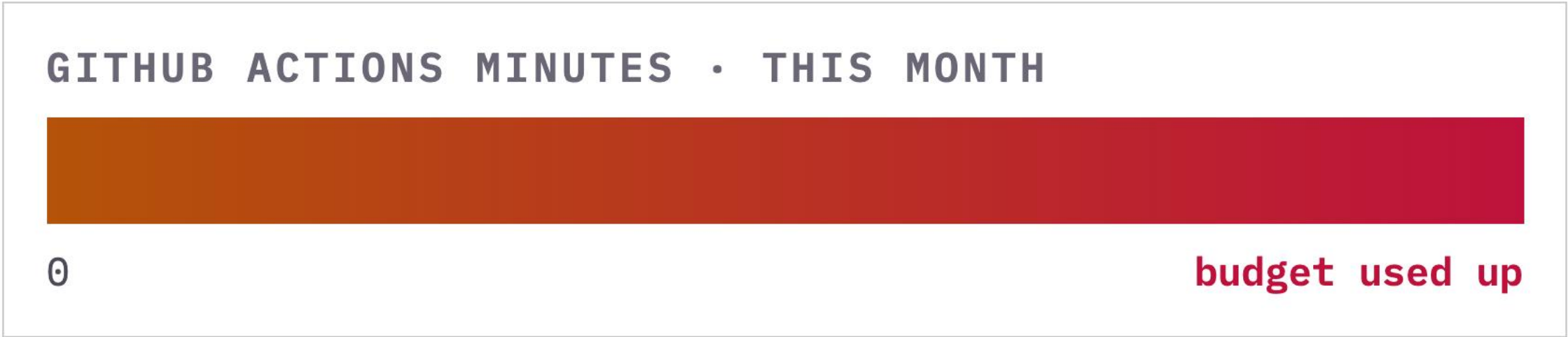
Final status	merged
PROBLEM Reports lived only on local disk; a sandbox run lost them.	
SOLUTION A reports module: store, list, review flag, findings ladder.	
KEY DECISIONS · WHY · SOURCE One row per run + kind retries stay idempotent FR-RP-1 HTML never inline in JSON keeps it sandboxed FR-RP-3 409 with a stable code clients branch on code rule: errors	
CONTEXT MANIFEST · WHAT THE AGENT READ #120 spec server/rules/coding.md server/rules/testing.md design-principles.md reports/service.ts	
VERIFICATION bun test reports · type-check clean · Closes #121	

THE WHY LEDGER

Presentation layer never writes #2410 · INT-1
Cookie door, no bearer in browser #2398 · FR-HC-1
Reports idempotent per run #2415 · FR-RP-1

Decisions are traced to a requirement or a rule, never invented. You can audit them months later.

Fix one bottleneck, meet the next



The codebase is just one kind of content.

```
company/  
├─ sales/      playbook • deals  
├─ marketing/  positioning  
├─ product/    specs • research  
├─ finance/    model • pricing  
├─ legal/      contracts  
└─ strategy/   decisions
```

Navigational

where is what

Operational

who does what

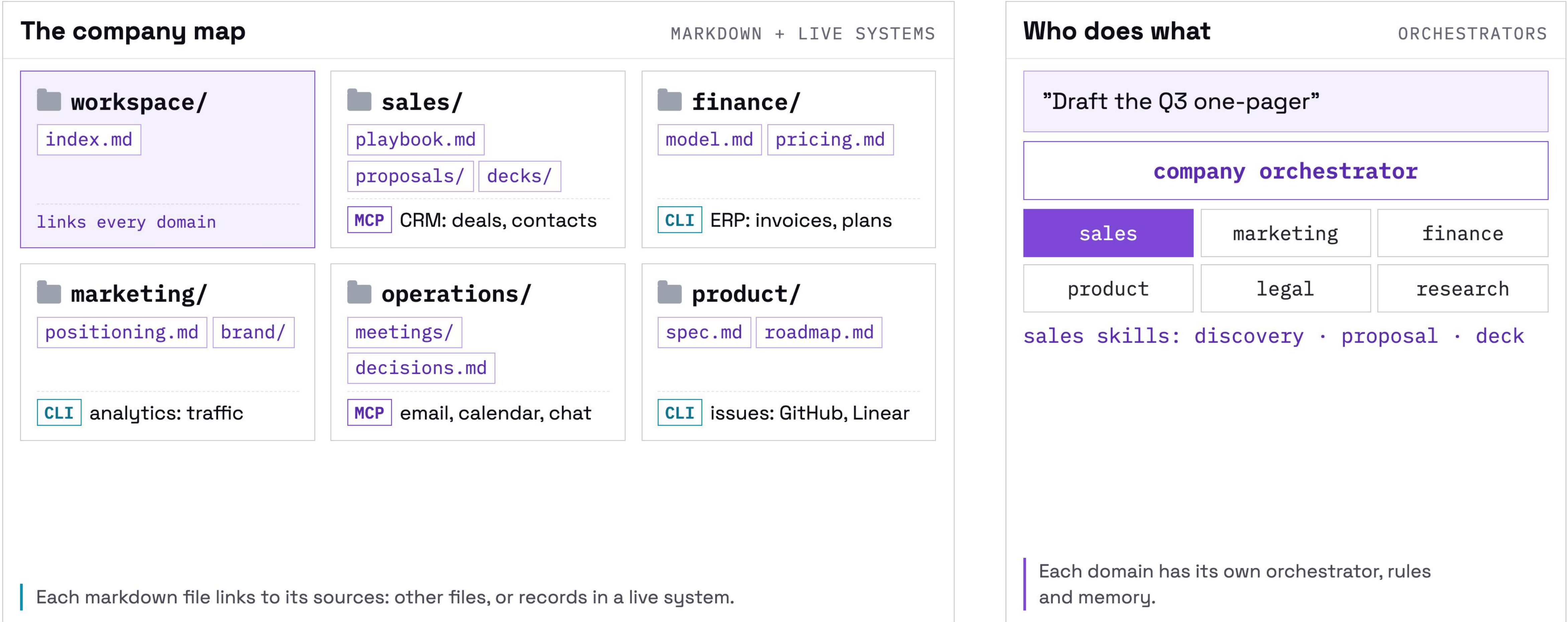
Content

what they work on

Dynamic context

how we do it, what we learned

The company works like the codebase





Burak Özafsar

CTO · MUTAGENT

Building the Agent Factory
at Mutagent.

burak@mutagent.io

**See it work on
one of your own
agents.**

